

**METHOD AND SYSTEM FOR ALGORITHM SYNTHESIS  
IN PROBLEM SOLVING**

5

**CROSS REFERENCE TO RELATED APPLICATIONS**

~~The following copending applications, Attorney Docket Number D/A1215,~~  
~~U.S. Application Serial No. <sup>Patent 6,865,562</sup> <sup>patent 6,161,011</sup>, filed 6/1/01, titled "Adaptive Constraint Problem Solving~~  
Method and System", is assigned to the same assignee of the present application. The  
entire disclosure of this ~~copending application~~ is totally incorporated herein by  
reference in its entirety.

**INCORPORATION BY REFERENCE**

The following U.S. patents are fully incorporated herein by reference: U.S.  
Pat. No. 5,871,731 (Trif et al., "Adaptive Problem Solving Method and System");  
15 U.S. Pat. No. 6,088,690 (Gounares et al., "Method and Apparatus for Adaptively  
Solving Sequential Problems in a Target System Utilizing Evolutionary Computation  
Techniques"); and U.S. Pat. No. 6,144,953 (Sorrells et al., "Time-Constrained  
Inference Strategy for Real-Time Expert Systems").

20

**BACKGROUND OF THE INVENTION**

This invention relates generally to the field of computerized problem solving  
and in particular to a system and method utilizing an adaptive constraint solver for the  
solution of complex problems in real time, based on predictive complexity models and  
interleaving of different problem solving algorithms.

25

510  
received from solver model ~~code 590~~, are sent to step 530, which sends the problem P and the parameters C to step 550 within solving module ~~code~~ 520. For the purpose of illustration, parameters C may specify, among other decision points, how many individual search steps the search method in solver iteration step 550 should execute.

- 5 The complexity-directed fine-grained, interleaved algorithm synthesis module code of the problem solver resides within solver iteration step 550, but for clarity is not shown. Other types of parameters and heuristics C are possible, as will be appreciated by one skilled in the art. At step 550 the solver ~~code~~ attempts to solve the problem P. In this step, the solver code typically searches part of the search space, by using, for example,
- 10 a gradient-descent, Nelder-Mead, interior-point, depth-first search, or any other technique that is intended to lead the solver towards promising regions of the search space, those regions where optimal valid solutions may be found. Step 550 depends on the type of solver or solvers used, as would be appreciated by one skilled in the art.

- Once step 550 has been performed, step 560 checks the result to determine if a
- 15 solution has been found, which is the case if no constraints are violated by the values of the solution and if no improvement in the objective function is found. If a valid solution has not been found, then the actual problem solver behavior is routed to step 570, where the performance so far is reviewed. Here, a decision is made as to whether another solver iteration step 550 should be executed, or whether the current status of
- 20 the problem and the behavior  $B_a$  should be sent to the complexity module ~~code~~ for an adaptation step. For example, the solving module ~~code~~ may have a fixed policy of running at least a minimum number of iterations before asking for adaptation. The actual behavior  $B_a$  and expected behavior  $B_e$  are compared at step 540 and resulting behavior  $\Delta B$  is routed back to step 530. If the actual behavior  $B_a$  and the expected
- 25 behavior  $B_e$  are different, parameters C are modified at step 530. For example, if the rate of improvement of the objective (specified in  $B_a$ ) is lower than expected (specified in  $B_e$ ), the number of individual search steps specified in C may be increased. The above process is then repeated until a solution is found.